DOCKET NO. BREITBART 14-8-1-39 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

RECEIVED

DEC 2 7 2002

Technology Center 2100

Commissioner for Patents
Washington, D. C. 20231

> I hereby certify that this correspondence is being deposited with the United States Postal Service as first
> class mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on
> 12/18/02 (Date)
> Judy A. Naon
> (Printed or typed name of person signing the certificate)
> (Signature of the person signing the certificate)

Sir:

LETTER TO OFFICIAL DRAFTSMAN

Transmitted herewith are five sheets of formal drawings to be substituted for the informal

drawings initially filed in the above-identified application for patent.

Respectfully submitted,

HITT GAINES & BOISBRUN, P.C.

David H. Hitt
Registration No. 33,182

Date: DECEMBER 17, 2002

P.O. Box 832570
Richardson, Texas 75083
(972) 480-8800

1/5

FIG. 1

FIG. 2
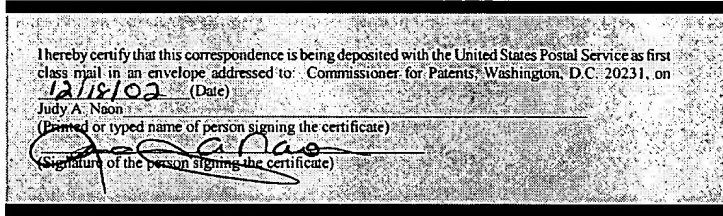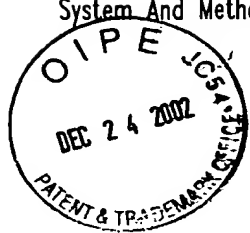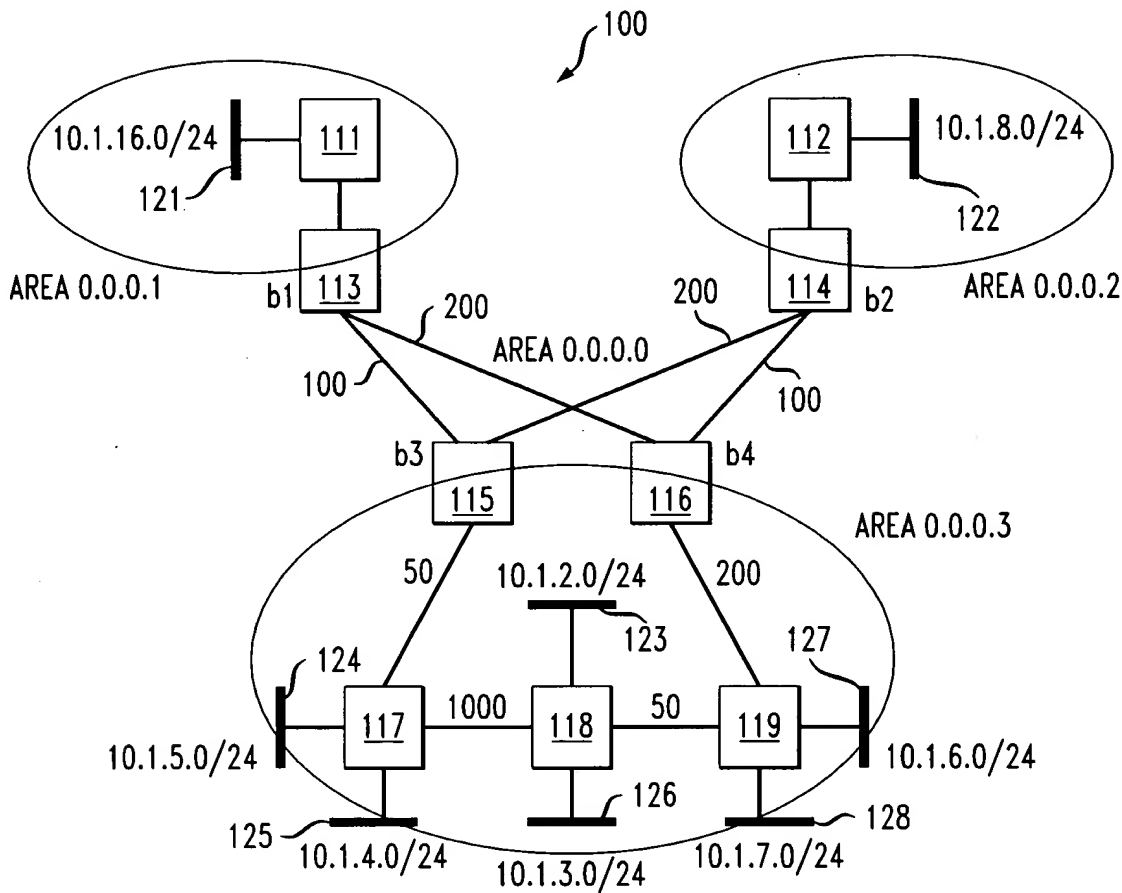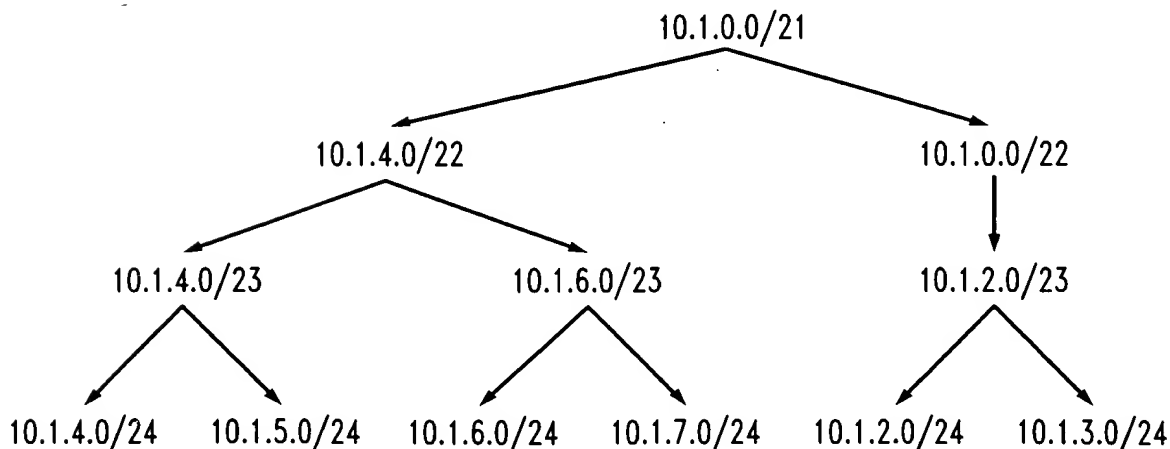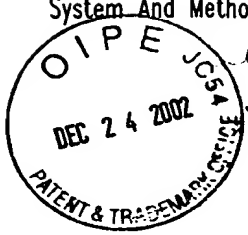
System And Method For Optimizing Open Shortest Path First Aggregates And Autonomous Network Domain Incorporating The Same
Breitbart 14-8-1-39
Serial No.: 09/775,329
Hitt, Gaines & Boisbrun, PC; D. H. Hitt (972) 480-8800

2/5

FIG. 3

```
procedure COMPUTEMINERROR(Aggregate x, Aggregate y, integer l)
1.  if subTree[x, y, l].computed = true
2.     return [subTree[x, y, l].error, subTree[x, y, l].aggregates]
3.     minError := minError1 := minError2 := ∞
4.  if x is a leaf {
5.     minError1 := Σ_{s∈S} D(s, t) * (lsp(s, x, {y}, W_A) - lsp(s, x))
6.     if l > 0
7.        minError2 := Σ_{s∈S} D(s, t) * (lsp(s, x, {x}, W_A) - lsp(s, x))
8.     if minError1 ≤ minError2
9.        [subTree[x, y, l].error, subTree[x, y, l].aggregates] := [minError1, Ø]
10.    else
11.       [subTree[x, y, l].error, subTree[x, y, l].aggregates] := [minError2, {x}]
12.}
13. if x has a single child u {
14.    [minError1, aggregates1] := COMPUTEMINERROR(u, y, l)
15.    if l > 0
16.       [minError2, aggregates2] := COMPUTEMINERROR(u, x, l - 1)
17.    if minError1 ≤ minError2
18.       [subTree[x, y, l].error, subTree[x, y, l].aggregates] := [minError1, aggregates1]
19.    else
20.       [subTree[x, y, l].error, subTree[x, y, l].aggregates] := [minError2, aggregates2 ∪ {x}]
21.}
22. if x has children u and v {
23.    for i := 0 to l {
24.       [minError1, aggregates1] := COMPUTEMINERROR(u, y, i)
25.       [minError2, aggregates2] := COMPUTEMINERROR(v, y, k - i)
26.       if minError1 + minError2 < minError
27.          minError := minError1 + minError2
28.          aggregates := aggregates1 ∪ aggregates2
29.    }
30.    for i := 0 to l - 1 {
31.       [minError1, aggregates1] := COMPUTEMINERROR(u, x, i)
32.       [minError2, aggregates2] := COMPUTEMINERROR(v, x, k - i - 1)
33.       if minError1 + minError2 < minError
34.          minError := minError1 + minError2
35.          aggregates := aggregates1 ∪ aggregates2 ∪ {x}
36.    }
37.    [subTree[x, y, l].error, subTree[x, y, l].aggregates] := [minError, aggregates]
38.}
39. subTree[x, y, l].computed := true
40. return [subTree[x, y, l].error, subTree[x, y, l].aggregates]
```

System And Method For Optimizing Open Shortest Path First Aggregates And Autonomous Network Domain Incorporating The Same
Breitbart 14-8-1-39
Serial No.: 09/775,329
Hitt, Gaines & Boisbrun, PC; D. H. Hitt (972) 480-8800

3/5

FIG. 4

```
procedure COMBINEMINERROR()
1.   for i = 1 to m
2.     for j = 0 to k {
3.        Ti[j].[error, aggregates] := COMPUTEMINERROR(r(Ti), ∈, j)
4.        Xi[j].[error, aggregates] := [ ∞, Ø]
5.     }
6.   for j = 0 to k
7.     X1[j].[error, aggregates] := T1[j].[error, aggregates]
8.   for i = 1 to m
9.     for j = 0 to k
10.      for l = 0 to j
11.        if (Xi-1[l].error + Ti[j - l].error < Xi[j].error) {
12.          Xi[j].error = Xi-1 [l].error + Ti[j - l].error
13.          Xi[j].aggregates = Xi-1[l].aggregates ∪ Ti[j - l].aggregates
14.        }
```
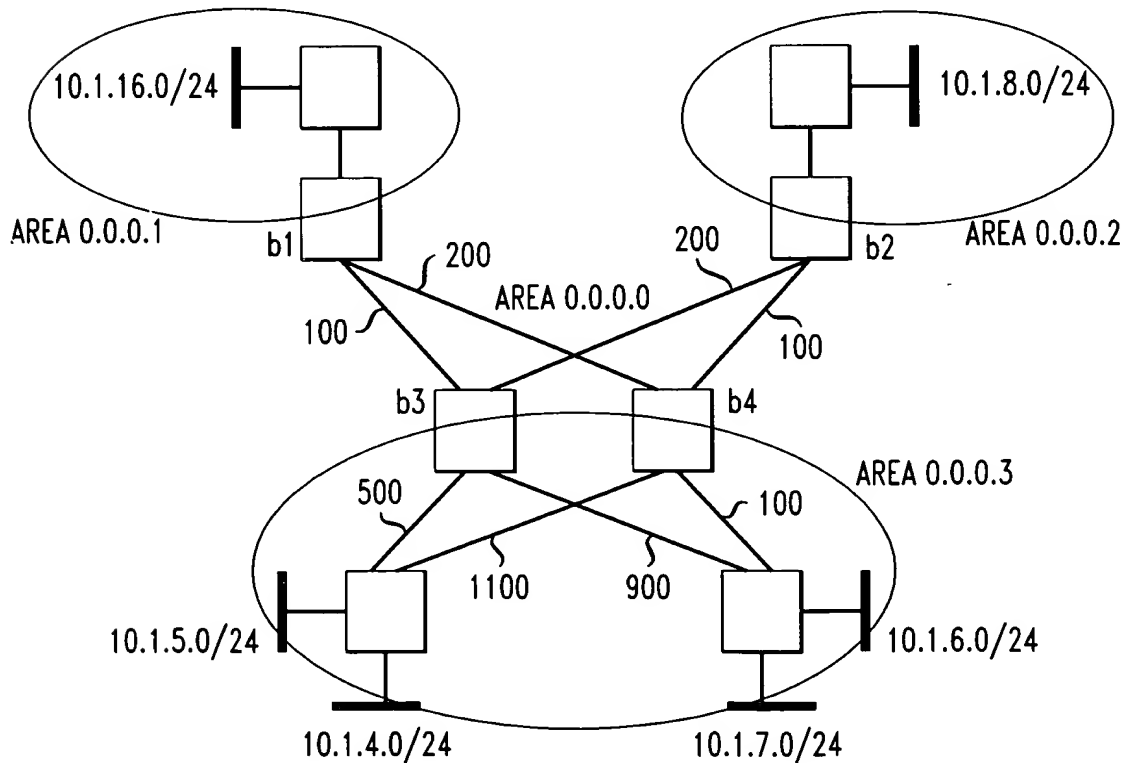
FIG. 5

4/5

FIG. 6

s_{xi} $s_{\overline{xi}}$

lsp=w+d1
e=1

lsp=w-d2
e=1

lsp=w
e=0

lsp=w
e=0

b_{xi} $b_{\overline{xi}}$

FIG. 7A

s_{Cj} s_{xi}

lsp=w-d4
e=100

lsp=w+d3
e=100

lsp=w
e=0

lsp=w
e=0

b_{Cjl} (a) $C_{jl} = x_i$ b_{xi}

FIG. 7B

$s_{\overline{xi}}$ s_{Cj}

lsp=w-d4
e=100

lsp=w+d3
e=100

lsp=w
e=0

lsp=w
e=0

$b_{\overline{xi}}$ (b) $C_{jl} = \bar{x}_i$ b_{Cjl}

System And Method For Optimizing Open Shortest Path First Aggregates And Autonomous Network Domain Incorporating The Same
Breitbart 14-8-1-39
Serial No.: 09/775,329
Hitt, Gaines & Boisbrun, PC; D. H. Hitt (972) 480-8800

5/5
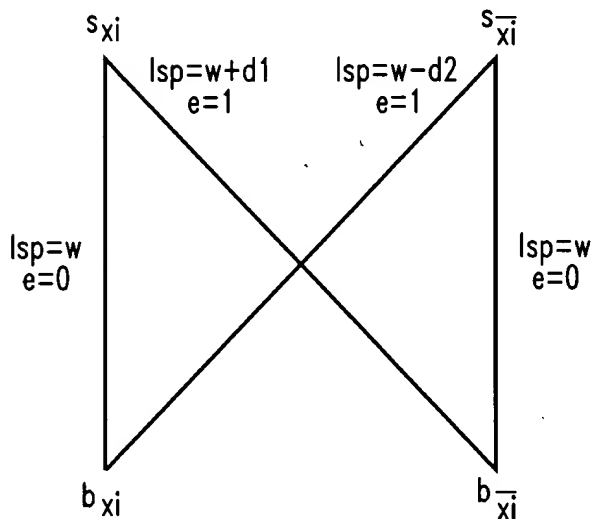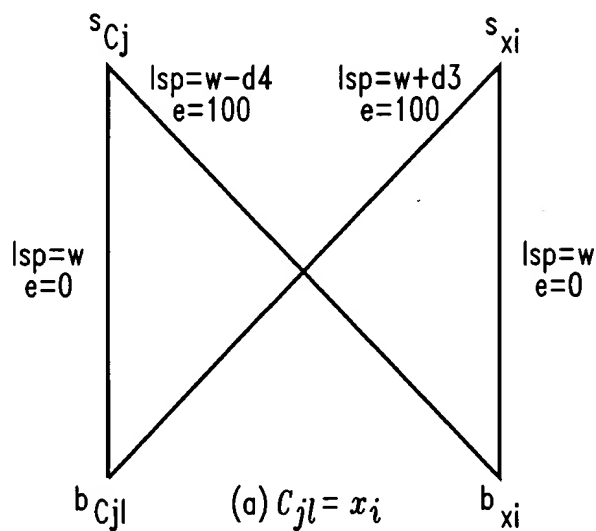
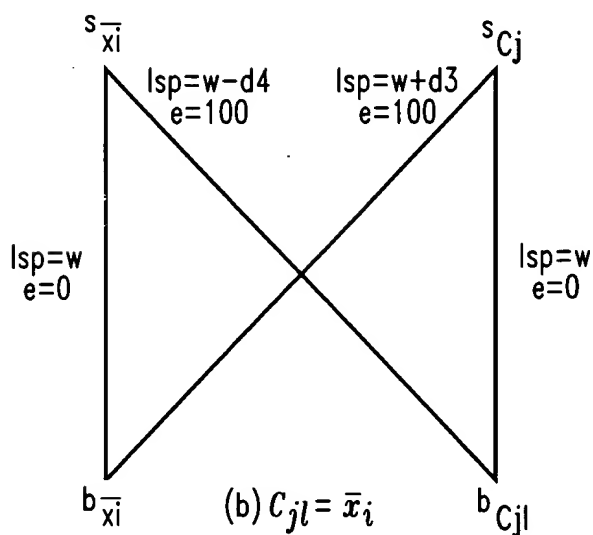*FIG. 8*

```
procedure COMPUTEWEIGHTSCUMULATIVE ()
1.   for each b ∈ Bᵢ set Wₘᵢₙ(b) := 0
2.   for i :=1 to r {
3.       W := Wₘᵢₙ
4.       Choose a random subset R ⊆ Bᵢ of ABRs
5.       for each b ∈ R set W(b) to a random weight in [0, L]
6.       if Σₛ∈ₛe(s, B(s, W)) < Σₛ∈ₛe (s, B(s, Wₘᵢₙ))
7.           Wₘᵢₙ := W
8.   }
9.   return Wₘᵢₙ
```

FIG. 9

```
procedure ComputeWeightsMax(Q)
    1.for each b ∈ Bᵢ set Wold(b) := 0
    2. while (Pb₂B

i Wold(b) ≤ (

j Bᵢ j*(j Bᵢ j-1)

2 ) *lspmax) f3. Let
```

Q0 be a new set of inequalities that result when the value Wold(b) is substituted for each variable W (b)only on the LHS of each inequality in Q 4. Set Wnew(b) to the smallest possible value such that each inequality in Q0 is satisfied when Wnew(b) is substituted for variable W (b) in Q0 5. if Wnew= Wold 6. return Wnew 7. else 8. Wold := Wnew 9.g 10. return "there does not exist a weight assignment W"

FIG. 10

```
procedure COMPUTEWEIGHTSTWOABR()
1.   Set V_opt := v(s₁), E := E_opt := Σₛ∈ₛe(s, b₁)
2.   for j:=1 to n {
3.       E := E + e(sⱼ, b₂) - e(sⱼ, b₁)
4.       if E < E_opt
5.           V_opt := v(sⱼ₊₁), E_opt := E
6.   }
7.   return V_opt
```